

Interaction Styles

[Preece, Chap 13-16; Hix, Chap 3; Nielsen, Chap 3]

What do interfaces (and consequently, interaction techniques) do? (Foley et al. 1984)

- Selection: Make a selection from a number of alternatives
- Position: Indicate a position on the display (or in the workspace or environment)
- Orientation: Alter the orientation of an object in the workspace
- Path: Generate a path, which is a series of positions and orientations over time
- Quantify: Specify a value (i.e. a number) to quantify a measure
- Text: Input a text string

Dialogue Design Styles

- Question and Answer
- Form Fill-In
- Command Language
- Natural Language
- Menu Selection
- Direct Manipulation
- Windows (GUI)
- Multimedia
- Virtual Environments
- Personal Data Assistants
- Others?

Important Issues

- Initiative — Who has control?
- Training requirements
- Learning time
- Speed of use
- Generality/Flexibility/Power
- Ability to show current/default values
- Typing skill requirements
- Expression
- Cost

Question and Answer

Q & A Example (See Figure 8.)

Q & A Pros and Cons

- + Little training needed
- + Oriented toward naive users
- + Default or current values can be shown
- + Alternative answers can be indicated
- Slow for experienced users

Use BACK to back up and re-answer previous questions.
Use STOP to stop the questioning.

What is your marital status? (M=married, S=single)

> S

How many dependents do you have?

> 2

Give the first dependent's first name.

> James

Give the second dependent's first name.

> BACK

Give the first dependent's first name.

> STOP

Figure 8.

- Best for use when answers are constrained or predictable
- Usually requires typing skill
- User does not have context of knowing what question will be asked
- Difficult to use in a complex environment
- ? Guides user — but system is in control
- + Good with many branch points

Q & A Guidelines

- Give proper syntax to be used for answering
- Show all possible answers if specific coding are required
For example: (M=married, S=single)
- Provide defaults if they make sense
- Provide escape to correct earlier answer and to abort

Form Fill-In (Spreadsheets) Example (See Figure 9.)

Form Fill-In Pros and Cons

- + Little training needed
- + Fast throughput, especially if defaults shown
- + User sees “big picture” context
- + Default or current selection/value can be shown
- + More user initiative than question and answer
- Computer initiated
- Hard to indicate a large set of alternatives
- Limited power
- ? Typically can't adapt single form based on value entered into one field
- ? Oriented toward data entry as opposed to commands

Form Fill-In Guidelines

- Use a meaningful title.
- Make form look like paper equivalent if one exists.
- Consistency of data entry transactions — let user's expectation be realized.

Enter the requested data in the boxes below.
To advance one field, press the TAB key. To go back one field,
press the TAB key while holding down the SHIFT key.
Press the EXECUTE key when finished entering data.

TITLE:

X_AXIS_LABEL:

X_AXIS_MIN:

X_AXIS_MAX:

Y_AXIS_LABEL:

Y_AXIS_MIN:

Y_AXIS_MAX:

NUMBER_OF_BARS:

STACKED_BARS (Y/N)?

	LABEL	VALUE1	VALUE2	VALUE3	VALUE4	VALUE5	...
BAR1	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
BAR2	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
BAR3	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
BAR4	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
BAR5	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
:							
:							
:							

Figure 9.

- Minimize input actions by the user.
- Distinguish between optional and required entries.
- Display default values.
- Error correction for characters and fields.
- Distinguish between default values and those explicitly entered by the user.
- Explicitly show minimum and/or maximum length.
- Require a single keystroke or selection to move from one field to the next.
- Vertically align fields
- Do not require user to left or right justify entry.
- Do not force user to put in spaces or required text (e.g. () or - in phone numbers).

Command Language Example (See Figure 10.)

```
% mkdir my_junk
% mv file1 file2 my_junk
% rm file3
% rm -r my_junk
```

Create a Pie Chart

```
> pie 3
> values 5, 10, 15
> font times-roman
> size 18
> color yellow
> title "Pie Chart Demo"
> slice 1; green
> slice 2; yellow
> slice 3; red
```

Figure 10.

Command Languages Pros and Cons

- + User initiated
- + Very fast, especially for touch typists using abbreviations (or speech recognition)
- + Can be very powerful and general
- + Large set of commands readily accessible
- Large learning curve
- Heavy memory load
- Error prone
- ? Assume the use of a keyboard

Command Language Guidelines

- Provide a consistent syntax
- Allow simple abbreviations — have a simple rule
- Use mnemonic names that are phonetically distinguishable
- Avoid long parameter lists — add more commands instead
- Place optional parameters on the end (probably required by implementation)
- Avoid requiring frequent use of shifted or control characters
- If error occurs, require re-entry only of erroneous data fields (tsh versus csh or sh)
- Allow defaults to minimize typing load
- Allow type-ahead
- Allow macros

Natural Language Example (See Figure 11.)

- Interactive natural language-like dialogue with immediate command feedback
- Computer output is on upper-case lines
- User input is on lower-case and quoted lines

```

GENERATE LEVEL ... ENTER
create a monthly bar chart

GENERATING A MONTHLY BAR CHART

ENTER:
title of the chart is "SALES VOLUME BY DISTRICT"

TITLE TEXT IS "SALES VOLUME BY DISTRICT"

ENTER:
label on the x axis is "MONTHS OF 1996"

X AXIS LABEL TEXT IS "MONTHS OF 1996"

ENTER:
the label on the y axis is "MILLIONS"

Y AXIS LABEL TEXT IS "MILLIONS"

ENTER:
input data

"WEST"
3 49.7 63.8 5 77.9 6 68.3
"EAST"
3 64.9 4 68.7 5 62 6 58.7

end of data

ENTER:
go

```

Figure 11.

Natural Language Pros and Cons

- + Can be either user or computer initiated
- + Successful in limited domain applications
- + Basic concepts are easy to learn
- Natural language can be complex and ambiguous
- Hard to train users to constrain vocabulary and requests
- Requires good user typing (or speaking) and spelling skills or a very good voice recognition system

Menu Selection Example (See Figure 12.)

Menus: Pull-down versus Pop-up — What's the difference?

A pulldown menu is what you typically see from a menubar (like at the top of any Macintosh screen or Windows 95 window. It can be context sensitive based on the window's content or the state of the application. This is most often done by graying out invalid commands or, in extreme cases, altering the content of the menus. This is a dangerous thing to do, however. Users don't necessarily understand why the menu changed.

A pop-up menu is a menu that pops up based on spatial context. Most X Window managers make use of this technique. Within an xterm window, the third button on the mouse pops-up an xterm-specific menu. If the mouse

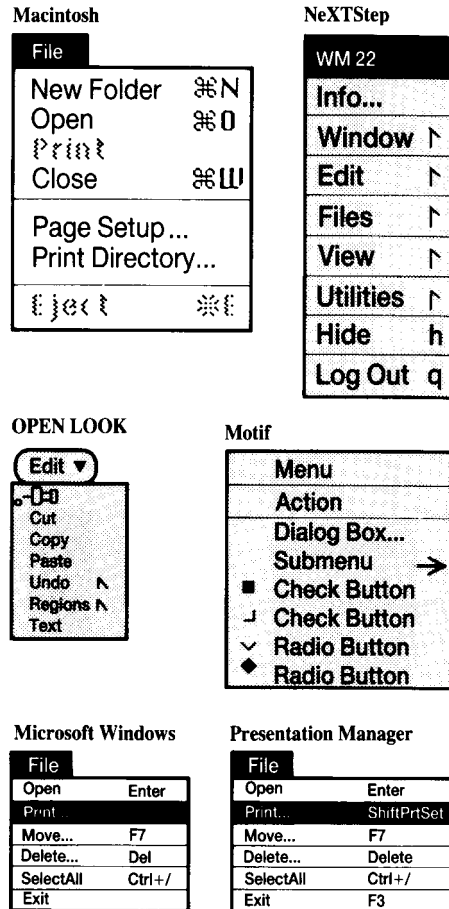


Figure 12.

moves outside the window to the desktop, the menu changes. This change of context can also be done within applications.

What's the difference between graphical menus (Pull-downs and Pop-ups) and non-graphical textual menus?

While it is true that graphical menus are an outgrowth from non-graphical, textual menus (e.g. Lotus 1-2-3), they are not functional equivalents.

- Graphical menus require a pointing device ...
- Non-graphical menus do not.
- Hierarchical graphical menus automatically show context (you can see how far down you are) ...
- Non-graphical menus only show the current submenu and must explicitly show context.
This problem is nearly identical to problems of becoming disoriented in hypertext (or web pages).
- Graphical menus require significant action (moving the mouse over large regions of screen space) ...
- Non-graphical menus require keypresses (typically)

The Pie Menu (Preece, pg266)

Similar to a pop-up menu, a pie menu is also spatially context sensitive but selections are made based on the direction of movement the mouse takes. When this is done in a hierarchical fashion, commands are selected based on a gesture (a series of directional movements) that can be learned with practice.

Menu Selection Pros and Cons

- + Very little training needed
- + Shows available options
- + Allows use of recognition memory
- + Hierarchy can expand selection (cascading menus)
- + Default or current selection can be shown
- Computer initiated
- Usable only with limited number of alternatives
- Slow for experienced users without express paths (a.k.a. accelerators)
- ? Most effective with pointing device

How many items to put in a menu?

Deep, narrow trees are the worst in terms of accuracy and speed.

Short, broad trees aren't much better.

Eight items with no more than two levels in a hierarchy seems to be near optimal.

Menu Structure (See Figure 13.)

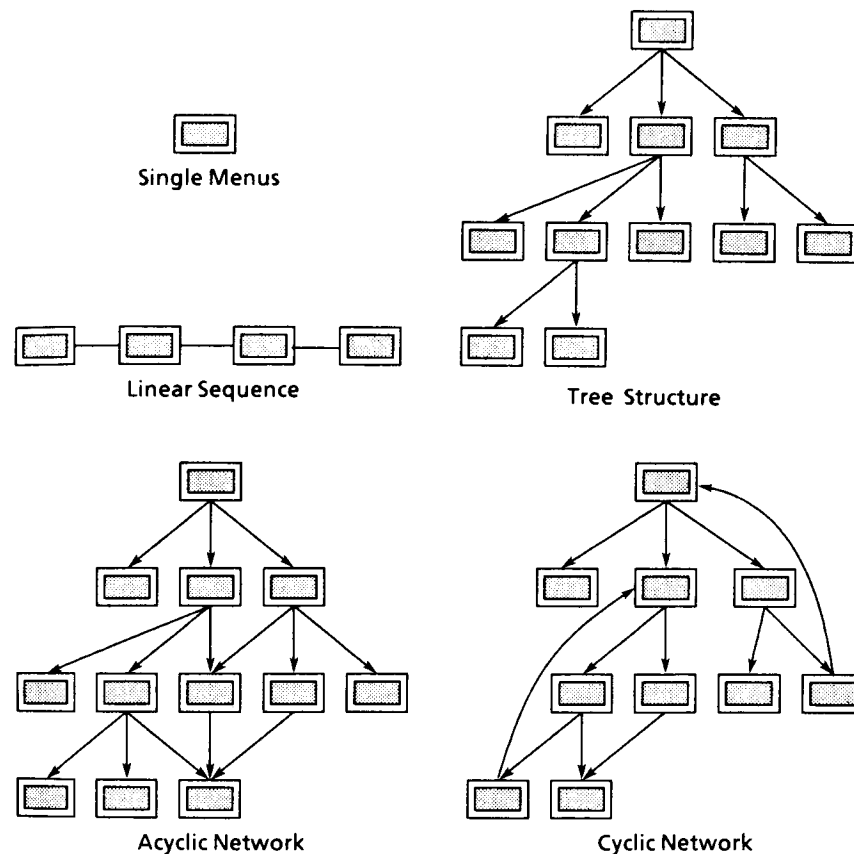


Figure 13.

Menu Selection Guidelines

- Use task semantics to organize menu structure
- Try to give position in organization by graphic design, numbering, and titles (especially true of non-graphical menus)
- Start with command/keyword if command language is also available
- Use alphabetic order for long lists if there is no clear semantic ordering
- Put frequently used alternatives first in short lists
- Group alternatives based on commonality; give each group a name or visual separation
- Show keyboard alternatives to menu selections if available (accelerators)
- Consider icons in place of or in addition to text — but don't overdo it.
- Make sure menu item groupings are non-overlapping
- Use familiar terminology

Direct Manipulation Definition

- Visual model of the world
- Visual objects are operated on
- Results of actions are reflected in the objects
- Actions are reversible and incremental
- Objects, once operated on, can be further operated on

Direct Manipulation Example (See Figure 14.)

Direct Manipulation Pros and Cons

- + User initiated
- + Easy to learn, intuitive
- + Fast to use with displayed objects
- + High retention by intermittent users
- + Easily augmented with menu selection and form fill-in
- + Provides closure of action and gesture, users see their progress (or lack thereof)
- Can be inconvenient and slow if user knows the name of undisplayed object but must find it anyway
- Limited power, not all desired actions have a “direct manipulation” analog (i.e. looking through a large file)
- Difficult to implement

Direct Manipulation Guidelines

- Make functionality clearly visible
- Follow the principle of least astonishment
- Use “magic” sparingly, and make its results clearly visible
- Objects that can be manipulated should be easily identifiable
- Actions on those objects should be equally obvious
- Use a simple metaphor if at all
- If a metaphor is used, don't violate it
- Use feedback to indicate valid or invalid actions

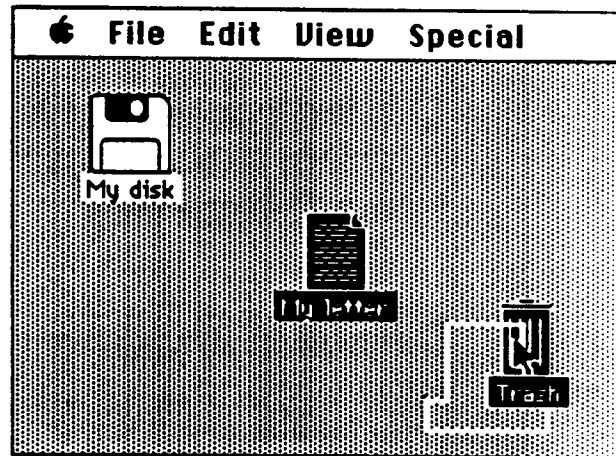
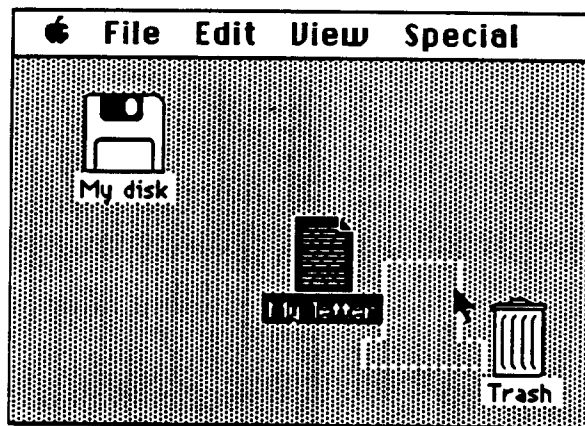
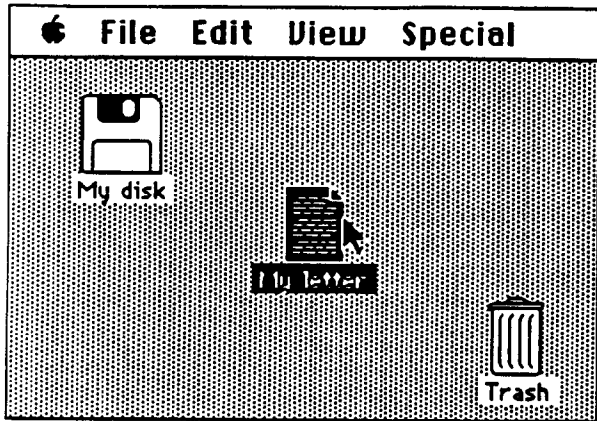


Figure 14.

Windows (GUI) Example

Often combined with direct manipulation interfaces.

Not all windows are alike. (See Figure 15., See Figure 16.)

Note the different locations for similar items on each window manager.

Not all controls are alike. (See Figure 17.)

Not all widgets are alike. (See Figure 19.)

Not all cursors are alike. (See Figure 19.) Does each mean the same thing? Ready? Wait? Delay?

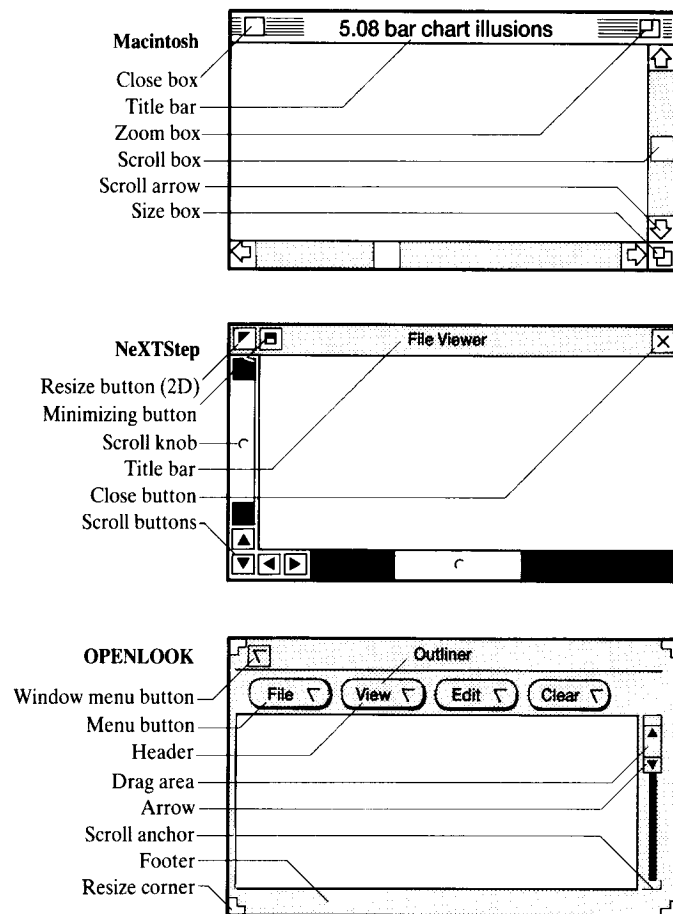


Figure 15.

GUI Pros and Cons

- + Makes extensive use of affordances
- + Easy to learn and use
- + Uses a metaphor
- Differences between systems are not always obvious (Windows 95 versus MacOS)

GUI Guidelines

- Don't overuse windows.
- Appearance and behavior of the primary window should be consistent.
- Use different windows for different independent tasks.
- Use different windows for different coordinated views of the same task.
- Use controls (widgets) that map to the task.

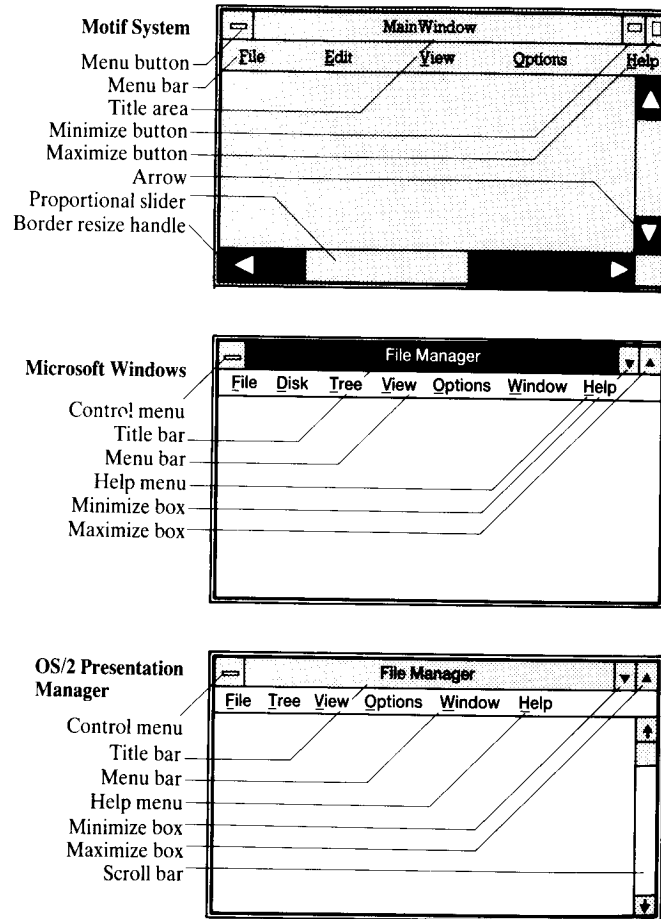


Figure 16.

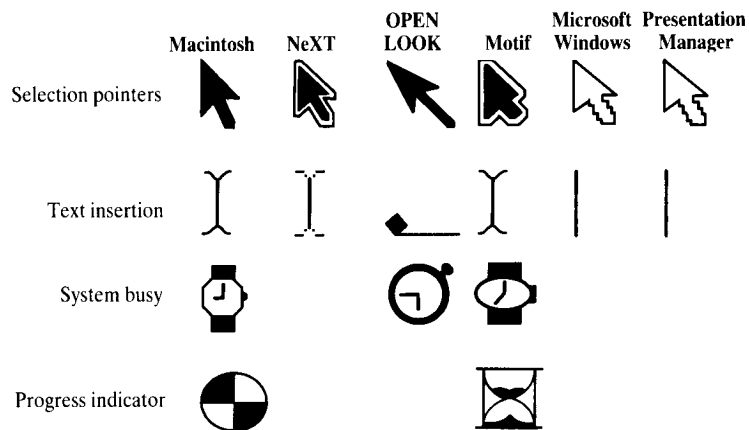


Figure 19.

Multimedia Example (See Figure 20.)

Virtual Environment Example (See Figure 21.)

Personal Data Assistant Example (See Figure 22.)

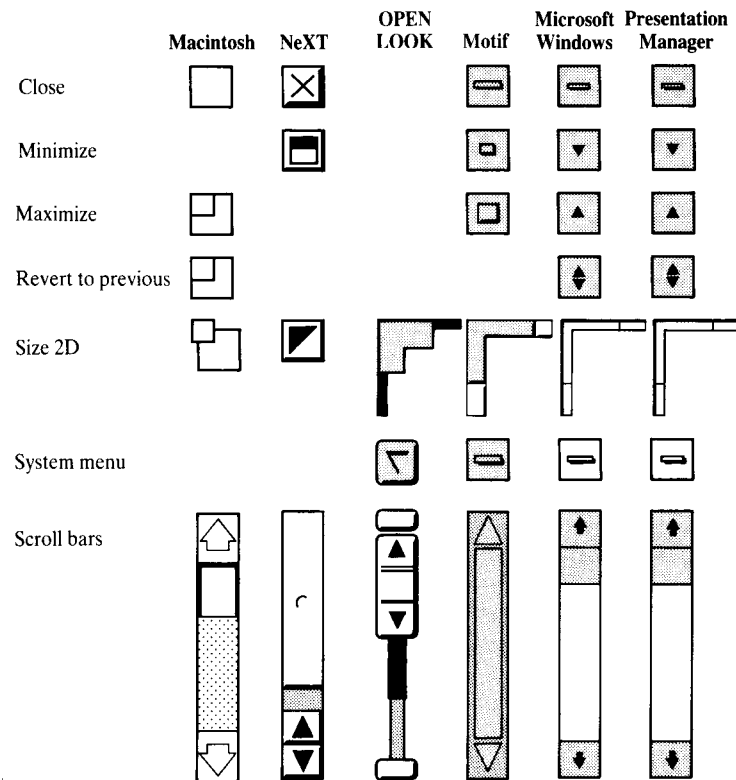


Figure 17.

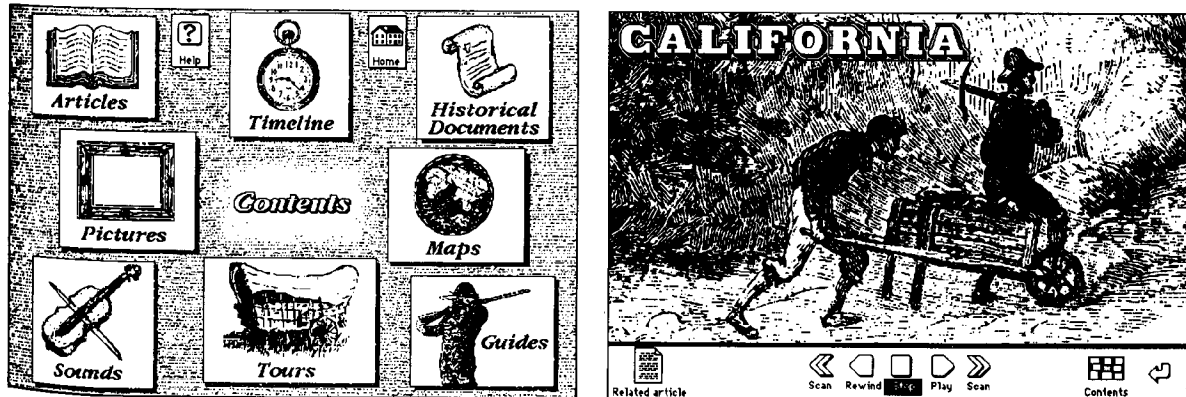


Figure 20.

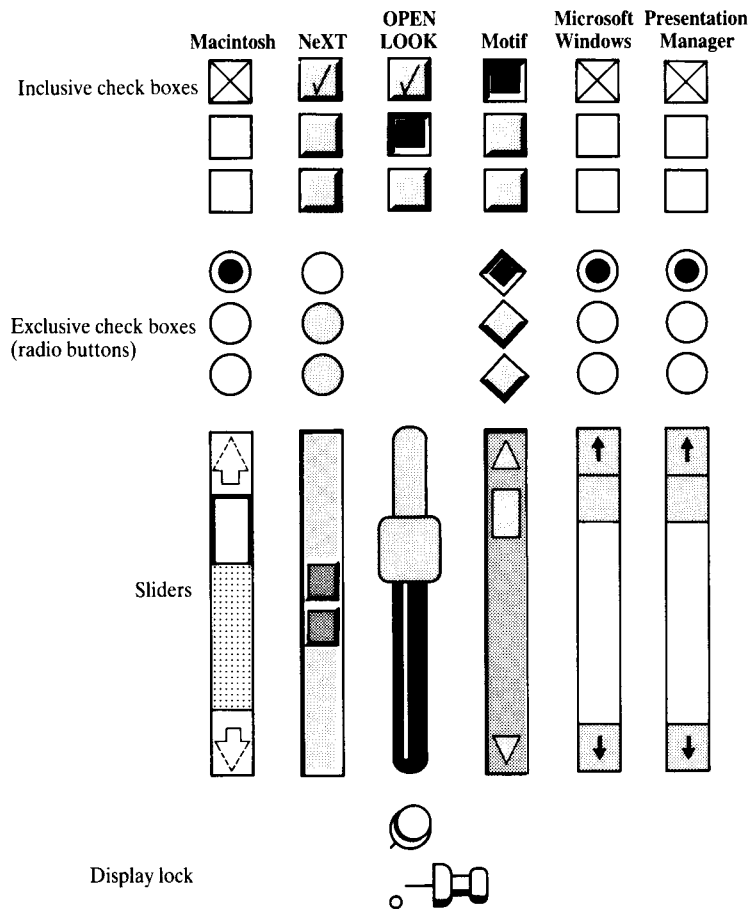


Figure 18.



Figure 22.

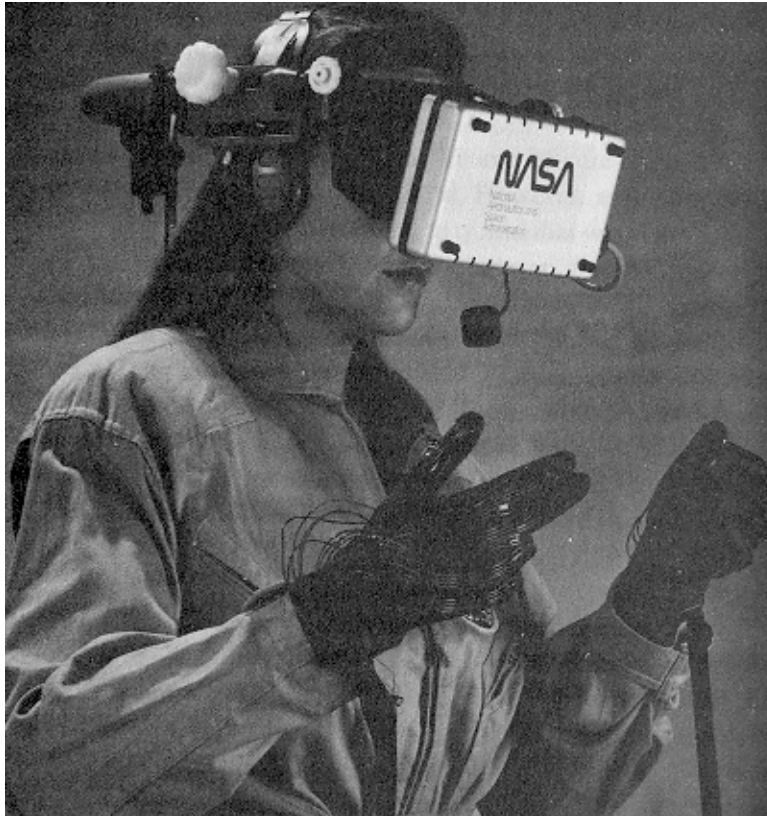


Figure 21.

	Direct Manipulation	Menu Selection	Form Fill-In	Command Language	Natural Language	Q & A
Learning Time	low	med	low	high	low	low
Speed of Use	med	med	high	high	med	low
Error Prone-ness	low	low	low	high	high	low
Extensibility	low	med	med	high	high	high
Typing Skill Required	none	none	high	high	high	high

Table 1: From Foley, vanDam, et al.